**Göğüs Kanseri Sınıflandırması**

**Classification Nedir?**

Classification (sınıflandırma), makine öğrenmesi algoritmalarının veriyi çeşitli kategorilere ayırmak için kullanılmasıdır. Spam filtreleme, el yazısı tanıma, hastalık teşhisi gibi alanlarda kullanılır. Algoritmalar test verileri ve sonuçları arasında ilişkiler kurup sonradan aldığı, daha önceden görmediği verileri bu ilişkilere dayanarak sınıflandırması mantığıyla çalışır.

**1- Veri Araştırması:**

Bu veri, Dr. William Wolberg'in 2,5 seneyi biraz aşkın klinik çalışmaları sonucu bir araya gelmiştir. Senede 2-3 defa güncellenen bu veri 15 Temmuz 1992'de çalışmalarda kullanılması için bağışlanmış olup güncellenme geçmişi şu şekildedir:

* **Grup 1:** 367 veri (Ocak 1989)
* **Grup 2:** 70 veri (Ekim 1989)
* **Grup 3:** 31 veri (Şubat 1990)
* **Grup 4:** 17 veri (Nisan 1990)
* **Grup 5:** 48 veri (Ağustos 1990)
* **Grup 6:** 49 veri (Güncellendi: Ocak 1991)
* **Grup 7:** 31 veri (Temmuz 1991)
* **Grup 8:** 86 veri (Kasım 1991)

Dr. Wolberg'in kendi tahlilleri üzerine dayandığı için güvenilir bir veri olarak değerlendirebiliriz. 2010'dan 2019'a kadar alıntılanmış olması da genel olarak güvenilir olduğunun bir göstergesi olarak sayılabilir.

Alıntılandığı makaleleri incelediğimizde veri biliminde kullanılan çeşitli algoritma ve metotların test ve incelemelerinde kullanıldığını görüyoruz. Bu türden çalışmalar için tercih edilmiş olmasında verinin nispeten pek de büyük olmaması ve genel olarak güvenilir bulunması etkili olmuş olabilir. Bunlar da bu veriyi bizim de güvenilir bulmamız için yeterli sebepler.

Özellikleri genel olarak incelersek:

* **Clump\_thickness:** Kümenin kalınlığı.
* **Uniformity\_of\_cell\_size:** Hücre boyutlarının düzenliliği.
* **Uniformity\_of\_cell\_shape:** Hücre şekillerinin düzenliliği.
* **Marginal\_adhesion:** Kanser hücrelerinin çevre dokulara bağlılık durumu.
* **Single\_epithelial\_cell\_size:** Epitel hücrelerinin boyutu.
* **Bare\_nuclei:** Hücre çekirdeklerinin durumu.
* **Bland\_chromatin:** Homojen veya düz chromatin dağılımı.
* **Normal\_nucleoli:** Nükleollerin morfolojik özelliği.
* **Mitoses:** Hücre bölünme süreci.
* **Class:** Kanserin türünü belirleyen, sınıf değeri.

Özelliklerin değer aralığı:

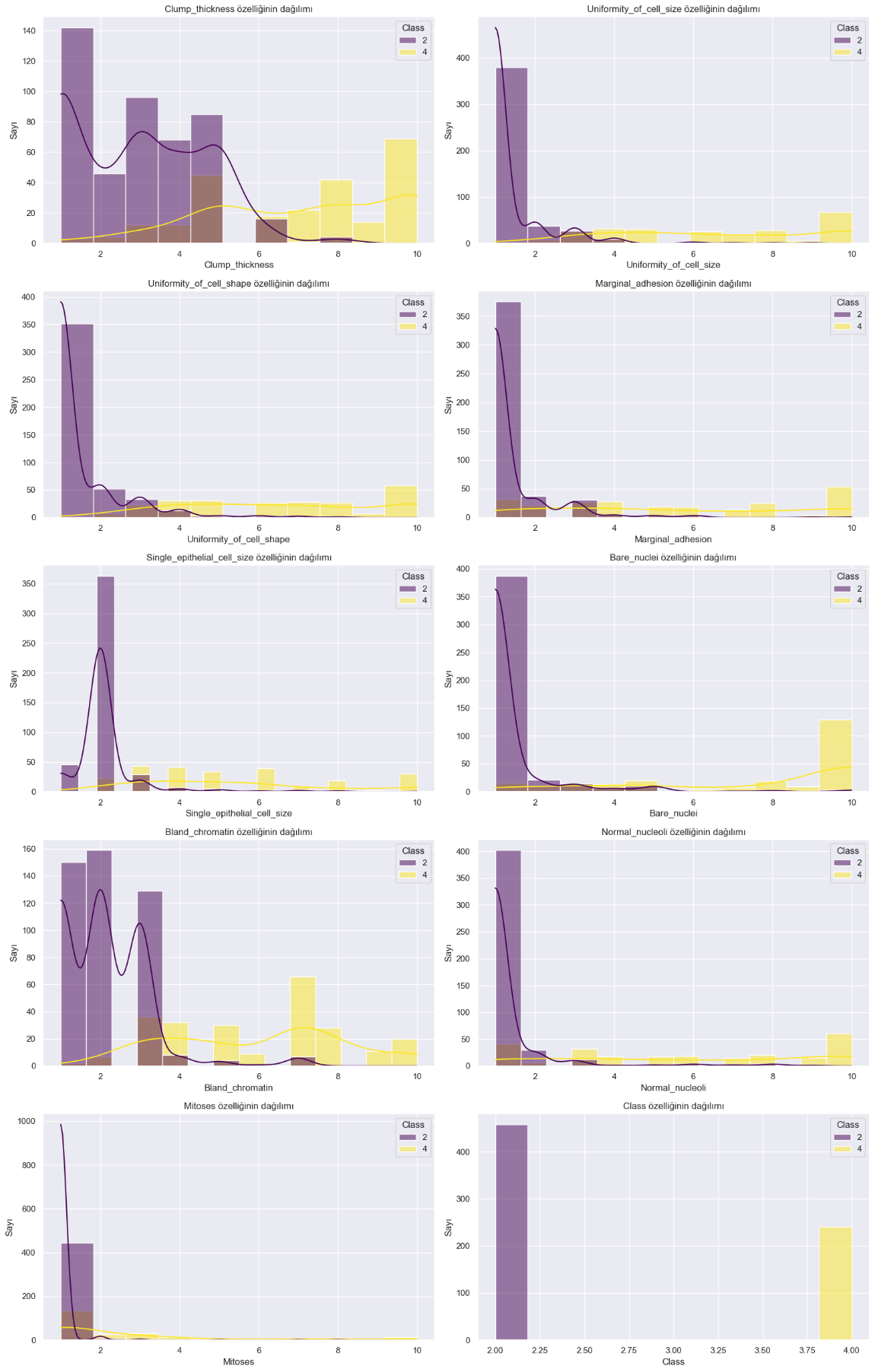
* **Clump\_thickness:** 1 – 10
* **Uniformity\_of\_cell\_size:** 1 – 10
* **Uniformity\_of\_cell\_shape:** 1 – 1
* **Marginal\_adhesion:** 1 – 1
* **Single\_epithelial\_cell\_size:** 1 – 10
* **Bare\_nuclei:** 1 – 10
* **Bland\_chromatin:** 1 – 10
* **Normal\_nucleoli:** 1 – 10
* **Mitoses:** 1 – 10
* **Class:** Kanserin türünü belirleyen, sınıf değeri.

Özelliklerin genel durumu:

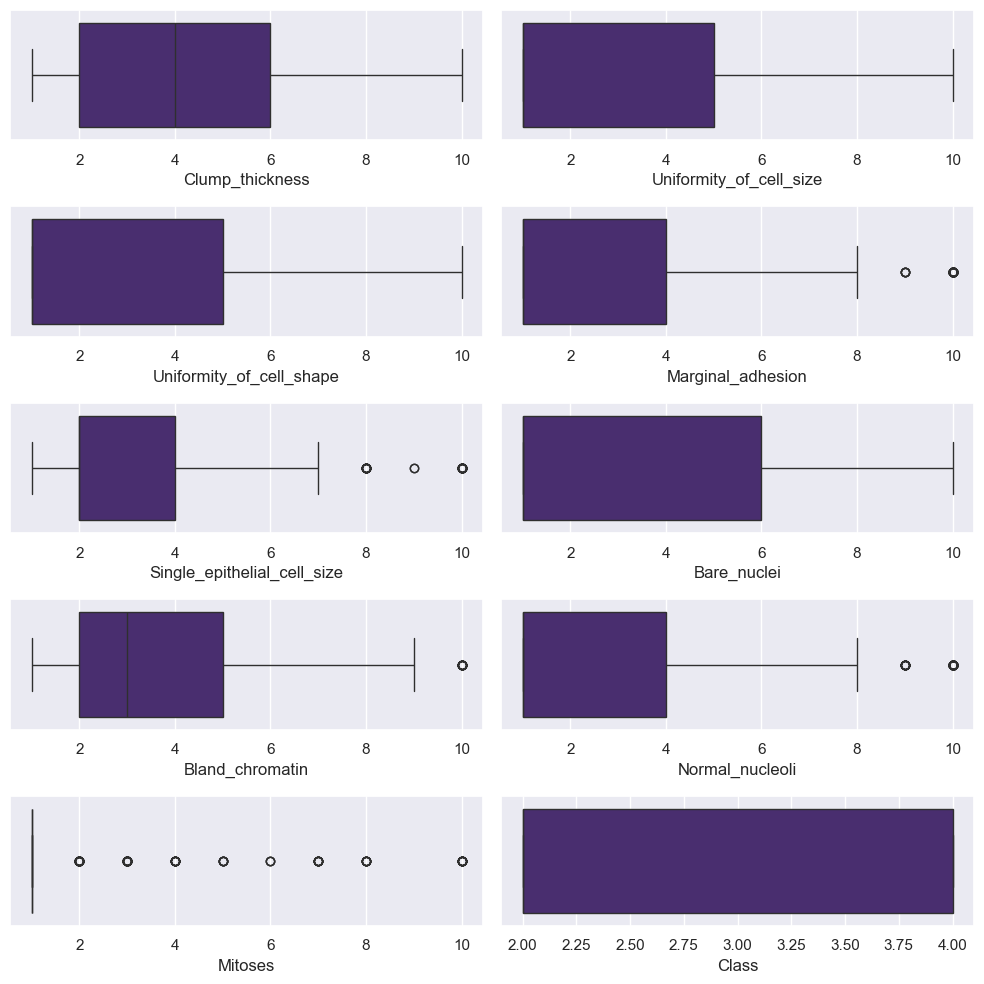
* **Clump\_thickness** 699 non-null int64
* **Uniformity\_of\_cell\_size** 699 non-null int64
* **Uniformity\_of\_cell\_shape**  699 non-null int64
* **Marginal\_adhesion**  699 non-null int64
* **Single\_epithelial\_cell\_size**  699 non-null int64
* **Bare\_nuclei**  683 non-null float64
* **Bland\_chromatin**  699 non-null int64
* **Normal\_nucleoli**  699 non-null int64
* **Mitoses**  699 non-null int64
* **Class** 699 non-null int64

Görebildiğimiz üzere Bare\_nuclei özelliğinde 16 adet NaN değer var.

Dağılım grafikleri:



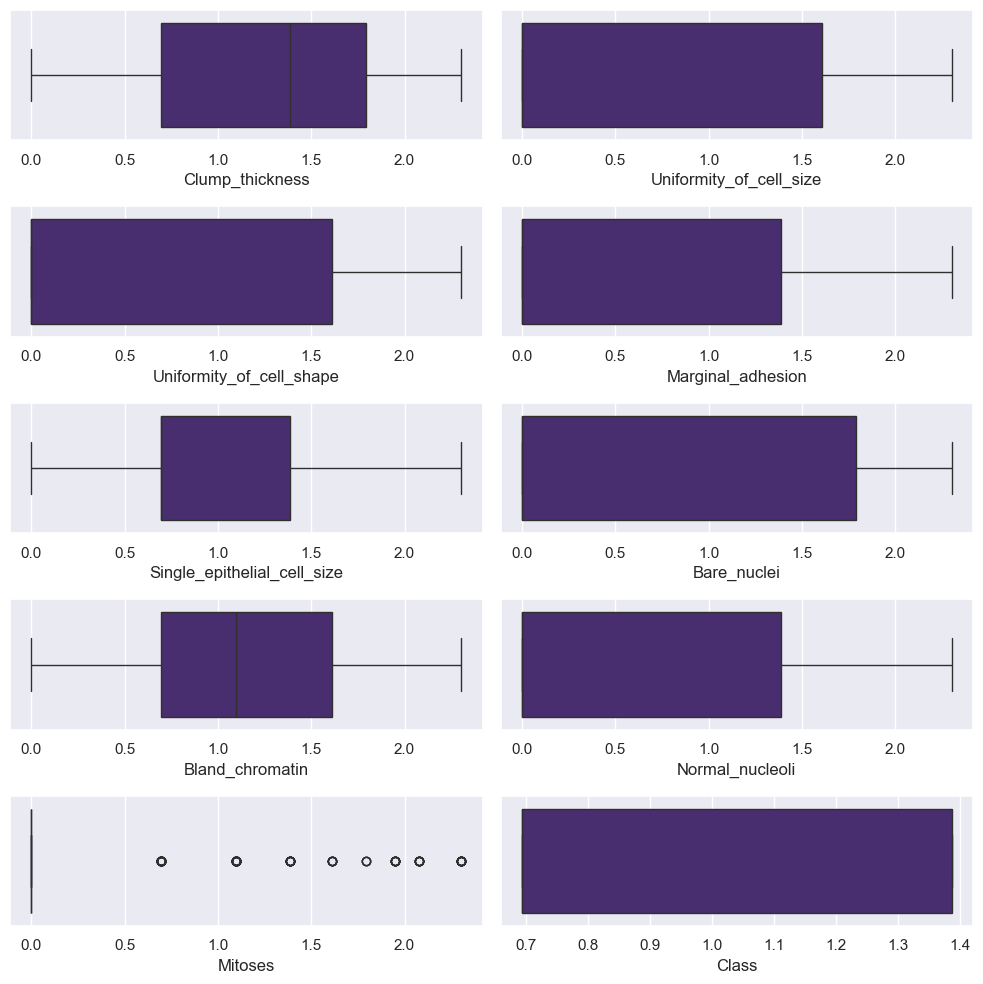
*Görsel 1: Özelliklerin genel dağılımı*

Boxplot olarak:

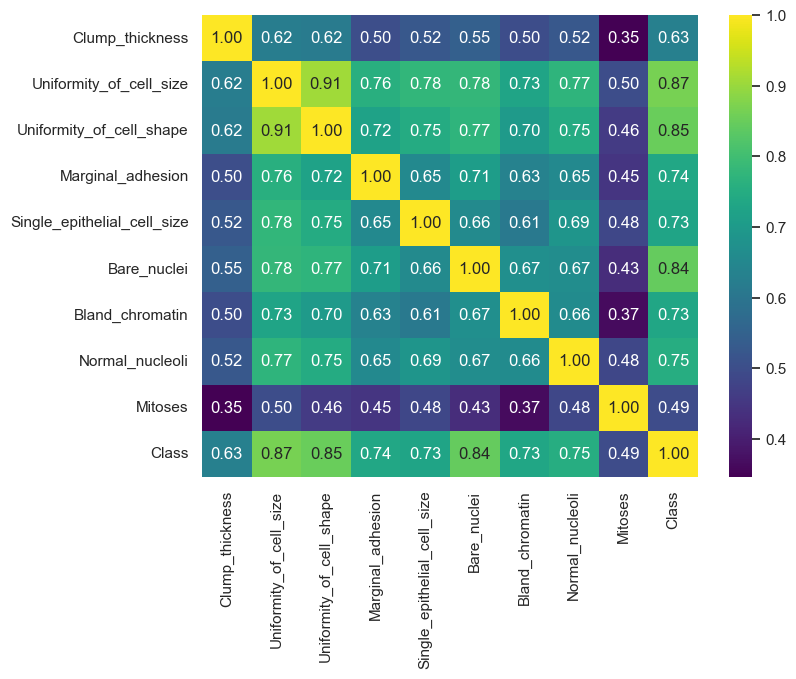
*Görsel 2: Boxplot dağılım gösterimi*

**2- EDA:**

Veriyi incelerken bazı NaN değerler olduğunu görmüştük. Bu değerleri atmak veya yerine yerleştirme yapmak gibi seçeneklerimiz var. Bu iki seçeneği de değerlendirebilmek için öncelikle veriyi NaN değerleri attığımız ve atmadığımız olarak ikiye ayırıyoruz.

Dağılım grafiklerini incelediğimizde bazı özelliklerin outlier bakımından kötü durumda olduğunu görüyoruz. Bunu biraz olsun düzeltmek adına log transformation uyguluyoruz. İşlem sonrası verinin yeni dağılımı:

*Görsel 3: Log transformation sonrası*

Bir miktar da olsa işe yaramış gibi göründüğü için NaN değerlerin atıldığı veriye de aynısını yapıyoruz.Sonraki işlem Class özelliğinin değerlerini yeniden kodlamak olacak. Başta 2 ve 4 olan bu değerleri genel olarak kullanılan 0 ve 1 ile değiştirerek daha anlaşılır hale getiriyoruz. Sonra özelliklerin birbirleriyle olan ilişkilerini incelemek için korelasyon matrisi oluşturuyoruz:

*Görsel 4: Korelasyon matrisi*

Sırada outlier kontrolü geliyor. Her ne kadar log transformation uygulamış olsak da veride hala outlier değerler bulunuyor. Boxplotların da gösterdiği üzere Mitoses özelliğinde outlierlar bulunuyor. Bu aşamada bu değerler için önümüzde üç seçenek bulunuyor: Bu değerleri veriden atmak, atmamak ve yerlerine veri yerleştirmek. Bu üç seçeneği de inceleyeceğiz, toplamda 6 farklı veri konfigürasyonu inceleyeceğiz yani. Yerlerine veri yerleştirmek için ileriye dönük bir adım olarak bu değerlerin yerine NaN yerleştirdiğimiz bir veri daha oluşturuyoruz.

Sonuç olarak outlierları atmadığımız, outlierları atmayıp başta NaN değerleri attığımız, outlierları attığımız, outlierları atmayıp başta NaN değerleri attığımız, outlierları NaN ile değiştirdiğimiz ve outlierları NaN ile değiştirip baştaki NaN değerleri attığımız 6 farklı veri bulunuyor elimizde. Modelleri test ederken NaN değerler için KNNImputer kullanıyoruz.

**3- Modellerin Kıyaslanması:**

Bu aşamada 6 veriyi ve bizden istenen 5 model ile bize bırakılmış bir modeli eşleştirip her ihtimali değerlendireceğiz. Ekstra model olarak GradientBoostingClassifier seçildi çünkü bu model özellikle overfittinge karşı mekanizmalara sahip olmasıyla biliniyor. Bunun potansiyel bir avantaj olabileceği düşünülerek bu model seçilmiştir.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | ROC AUC |
| XGBoost Classification | 0.9642857142857143 | 0.9545454545454546 | 0.9333333333333333 | 0.956140350877193 |
| Logistic Regression | 0.9714285714285714 | 0.9555555555555556 | 0.9555555555555556 | 0.9672514619883043 |
| Random Forest Classification | 0.9642857142857143 | 0.9545454545454546 | 0.9333333333333333 | 0.956140350877193 |
| Support Vector Classification | **0.9785714285714285** | **0.9565217391304348** | **0.9777777777777777** | **0.9783625730994152** |
| Neural Network Classification | 0.9428571428571428 | 0.9111111111111111 | 0.8444444444444444 | 0.9345029239766082 |
| Gradient Boosting Classification | 0.9571428571428572 | 0.9534883720930233 | 0.9111111111111111 | 0.9450292397660819 |

*Grafik 1: Outlierlar ve NaN değerler atılmamış veri*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | ROC AUC |
| XGBoost Classification | 0.9562043795620438 | 0.9814814814814815 | 0.9137931034482759 | 0.9505674378000872 |
| Logistic Regression | 0.9708029197080292 | 0.9821428571428571 | 0.9482758620689655 | 0.9678088171104322 |
| Random Forest Classification | 0.948905109489051 | 0.9636363636363636 | 0.9137931034482759 | 0.9442383238760367 |
| Support Vector Classification | **0.9854014598540146** | **0.9827586206896551** | **0.9827586206896551** | **0.9850501964207771** |
| Neural Network Classification | 0.948905109489051 | 0.9636363636363636 | 0.9137931034482759 | 0.9442383238760367 |
| Gradient Boosting Classification | 0.948905109489051 | 0.9636363636363636 | 0.9137931034482759 | 0.9442383238760367 |

*Grafik 2: Outlierlar atılmamış, NaN değerler atılmış veri*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | ROC AUC |
| XGBoost Classification | **0.9827586206896551** | **0.96** | 0.96 | 0.9745054945054945 |
| Logistic Regression | 0.9741379310344828 | 0.9583333333333334 | 0.92 | 0.9545054945054945 |
| Random Forest Classification | **0.9827586206896551** | **0.96** | 0.96 | 0.9745054945054945 |
| Support Vector Classification | **0.9827586206896551** | 0.9259259259259259 | **1.0** | **0.9890109890109889** |
| Neural Network Classification | **0.9827586206896551** | 0.9259259259259259 | **1.0** | **0.9890109890109889** |
| Gradient Boosting Classification | 0.9568965517241379 | 0.8846153846153846 | 0.92 | 0.9435164835164834 |

*Grafik 3: Outlierlar atılmış, NaN değerler atılmamış veri*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | ROC AUC |
| XGBoost Classification | **0.9734513274336283** | **1.0** | 0.8846153846153846 | 0.9423076923076923 |
| Logistic Regression | 0.9557522123893806 | **1.0** | 0.8076923076923077 | 0.9038461538461539 |
| Random Forest Classification | 0.9646017699115044 | 0.9583333333333334 | 0.8846153846153846 | 0.9365605658709107 |
| Support Vector Classification | **0.9734513274336283** | 0.96 | **0.9230769230769231** | **0.9557913351016799** |
| Neural Network Classification | **0.9734513274336283** | **1.0** | 0.8846153846153846 | 0.9423076923076923 |
| Gradient Boosting Classification | 0.9557522123893806 | 0.92 | 0.8846153846153846 | 0.9308134394341291 |

*Grafik 4: Outlierlar ve NaN değerler atılmış veri*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | ROC AUC |
| XGBoost Classification | 0.9571428571428572 | 0.9534883720930233 | 0.9111111111111111 | 0.9450292397660819 |
| Logistic Regression | 0.9571428571428572 | 0.9534883720930233 | 0.9111111111111111 | 0.9450292397660819 |
| Random Forest Classification | 0.9714285714285714 | 0.9555555555555556 | 0.9555555555555556 | 0.9672514619883043 |
| Support Vector Classification | **0.9857142857142858** | **0.9574468085106383** | **1.0** | **0.9894736842105264** |
| Neural Network Classification | 0.9571428571428572 | 0.9534883720930233 | 0.9111111111111111 | 0.9450292397660819 |
| Gradient Boosting Classification | 0.9571428571428572 | 0.9534883720930233 | 0.9111111111111111 | 0.9450292397660819 |

*Grafik 5: Outlierlar NaN ile değiştirilmiş, başta NaN değerler atılmamış veri*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | ROC AUC |
| XGBoost Classification | 0.9562043795620438 | 0.9814814814814815 | 0.9137931034482759 | 0.9505674378000872 |
| Logistic Regression | 0.948905109489051 | 0.9636363636363636 | 0.9137931034482759 | 0.9442383238760367 |
| Random Forest Classification | 0.9635036496350365 | 0.9818181818181818 | 0.9310344827586207 | 0.9591881274552597 |
| Support Vector Classification | **0.9708029197080292** | **0.9821428571428571** | **0.9482758620689655** | **0.9678088171104322** |
| Neural Network Classification | 0.948905109489051 | 0.9636363636363636 | 0.9137931034482759 | 0.9442383238760367 |
| Gradient Boosting Classification | **0.9708029197080292** | **0.9821428571428571** | **0.9482758620689655** | **0.9678088171104322** |

*Grafik 6: Outlierlar NaN ile değiştirilmiş, başta NaN değerler atılmış veri*

**Sonuç:**

Grafikleri ve kodun verdiği çıktıları genel olarak incelediğimizde SVC modelinin ezici üstünlüğe sahip olduğunu görüyoruz. Bu model bazında veri kıyaslaması yaptığımızda da 5. grafikte SVC modelinin en iyi değerine ulaştığını görüyoruz. Denediğimiz tüm kombinasyonlar arasında en iyi durum da budur diyebiliriz. Zira burada bir tıbbi teşhis koymaya çalışıyoruz. Bu durumda Recall, Precision’dan daha önemli bir metrik olacaktır. Zira yanlış bile olsa tüm pozitif teşhisleri doğru bilmek istiyoruz. Hasta olmayan birisine hasta olduğunu söylemek, hasta olan birisine hasta olmadığını söylemeye yeğdir. Dolayısıyla sonuç olarak, outlierları NaN ile değiştirip en son, baştaki NaN değerlerle beraber yerlerine veri koyduğumuz durumda SVC modeli hepsinin arasında en iyi sonucu vermiştir. Diğer verileri de kontrol ettiğimiz zaman SVC’nin asla berbat sonuç vermediğini görüyoruz. Buna dayanarak elimizdeki tüm farklı veriler için SVC’nin kullanılabilir bir model olduğuna karar verebiliriz.